# Topical Classification and Divergence on Reddit

**Amanda Chow**
amdchow@stanford.edu

**Jenny Hong**
jyunhong@stanford.edu

## Abstract

Online discussion forums are virtual spaces for communities of users to discuss various topics. As an online discussion progresses, there is a tendency for comments to diverge from the initial topic, often degenerating into off-topic shouting matches. We would like to determine how this divergence happens, and identify comments that trigger this topic drift. We approach this problem by first building a neural topic model, which we later use to analyze topic divergence at aggregate and individual thread levels.

## 1 Introduction

In most forums, online discussions (also called threads) are composed of comments, usually in a linear structure, or more generally, a tree structure, where users can post comments in reply to other comments. The deeper a comment is in a thread, the less likely it is to be about to the original topic.

We hypothesize that comments that start a discussion thread chain are the most on-topic, while later comments in the chain are more off-topic. We want to test this hypothesis and explore how this topic drift happens by answering some relevant questions:

- At an aggregate level, is there a general and measurable trend of topic drifting as we delve into deeper levels of conversation?
- In a particular comment thread, can we identify the comment that triggers a topic drift?

To study topic divergence at these levels, we break the problem down into two steps. First, we solve the task of classifying comments into topics via topic modeling. Then, we apply the classifier in various ways to study different levels of topic divergence, at aggregate- and thread-levels.

## 2 Background and Related Work

In general, no previous work has directly tackled the analysis of topic divergence in discussions. Discussion threads are commonly studied from the perspective of social dynamics [1]. The work perhaps most closely related to ours is on studying structure in discussion threads (e.g. discovering that one post is an answer to another post, which is a question) [8]. Unfortunately, neither of these directions combines the structure of a discussion forum with topical analysis.

Topic modeling itself has a wealth of previous work, which we use as a starting point for our classification task. These models include Term Frequency-Inverse Document Frequency (TF-IDF) [7], Latent Dirichlet Analysis (LDA) [3], and supervised LDA (sLDA) [2].

TF-IDF assigns a "weight" to each (word, document) pair, which increases with higher word count in the document and decreases with higher word count in the overall corpus (to account for frequent stop words). These pairs can be clustered to discover topics.

LDA models each comment as a mixture of topics that independently produce words with given probabilities, and uses the words in a comment to predict the topic that produced the comment with

the highest likelihood. The probability is $p(x^{(i)}|y^{(i)})$ of an instance $x^{(i)}$ coming from class $y^{(i)}$ is modeled as a Gaussian $N(\mu_{y^{(i)}}, \Sigma)$ with each class having its own mean but a shared covariance. LDA seeks to maximize $\sum_{i=1}^{N} p(x^{(i)}|y^{(i)})$ of all the data points.

sLDA is a supervised variation of LDA which modifies the maximum-likelihood estimator to include given labels.

These are popular and established topic models; however, they use word unigram and n-gram features, but not word embeddings, leaving room for a neural network approach to topic modeling. Previous work in word embeddings, such as GloVe [5], suggest that vector space models for words can carry semantic meaning beyond traditional tasks, such as binary sentiment analysis. We plan to use such embeddings in our project and see how well they perform in our context of multi-class classification.

## 3 Approach

Our study consists of two tasks. First, train a neural classifier to predict a class (subreddit, in this case) given the text of a comment. Then, we apply the classifier, first to a collection of comments at a certain "depth" of a discussion, then to specific threads.

### 3.1 Classification

The goal of our classifier is to predict the topic of a comment. We train the classifier on "top-level" comments, based on the assumption that top-level comments are close to a ground truth representation of what the topic of a forum is.

Before building a neural classifier, we first survey our dataset using popular topic models mentioned above. We use an unsupervised topic detection and compare the detected to topics to ours, to determine if our chosen labels are sufficiently distinct from each other. We also use supervised topic classifiers (e.g. with TF-IDF features) to set baseline metrics for classification. Then, for our classifier, we use a neural network whose architecture is detailed in Section 5.2.

### 3.2 Topical Divergence

As discussed in the introduction, we hope to use the classifier to answer a couple of questions about topical divergence at aggregate- and thread-levels.

**Aggregate-level.** We will first run our classifier on comments at different levels to look at aggregate behavior. The dataset is sampled from top-level comments that are not in the training data. We expect the classification accuracy to monotically decrease in deeper levels, and plan to plot this pattern. We can also look at the confusion matrices of the classifier at different levels. Again, we expect more "confusion" among classes at deeper levels.

**Thread-level.** Our next task is to look a little closer at individual threads. More specifically, we can map the comments of a thread into a topic space to determine when comments topically deviate. If a comment is identified as off-topic, we can measure how off-topic it is with a Kullback-Leibler (KL) divergence of its softmax predictions from the softmax predictions of the top-level comments. Ideally, these topic-changing comments will correspond with spikes in KL divergence.

## 4 Dataset

We use a popular discussion forum, Reddit, which is an online news and entertainment website. Reddit is divided into sub-forums called "subreddits," each focused on a specific theme or topic (e.g. `r/funny`, `r/GetMotivated`, `r/communism`). The posts in each subreddit are related to the subreddit's theme, and each post (along with its associated comments) belongs to exactly one subreddit. The comments on these Reddit posts are organized in a tree structure. This organization of Reddit into subreddits provides natural labels for topic classification.

We use the Reddit Comments Corpus [6] hosted on the Stanford Infolab servers. The corpus contains 1.65 billion posts from between October 2007 and May 2015, providing a comprehensive view of

Reddit. For each comment, the associated metadata includes the subreddit name, the text content, the parent comment, and other information.

We use 31 of the 50 default subreddits, omitting the largest subreddits. These generally do not have a specific, well-defined topic, such as `r/AskReddit` (a general question-answering forum). This still leaves us with some difficult subreddits to differentiate, such as `askscience` and `space`, or `InternetIsBeautiful` and `dataisbeautiful`.

### 4.1 Classifier Training Set

The topic (subreddit) classifier is trained on top-level comments (i.e. comments with no parent comment) of each post. As per our initial hypothesis, we assume that comments at deeper levels in threads sometimes diverge from the original topic. Our classifier should not capture these various other topics as part of a specific topic/subreddit. Therefore, these top-level comments are most representative of the subject of a subreddit, forming a high-quality, large, and labeled dataset for subreddit modeling. All comments are labeled with the subreddit they belong to, so the problem is a purely supervised one.

While training the neural topic model, we found ourselves limited by the computation power of machines we could access. Training on all top-level comments in the dataset, a single epoch took 3 days to complete. Thus, for the purposes of this project, we randomly sampled a subset of top-level comments to produce a "medium"-sized dataset for a total of 217,000 training examples and 62,000 validation examples. This cut our training time down to just over one hour for a single epoch. We also extracted the same number of training examples (7,000) and validation examples (2,000) for each of the 31 classes, to reduce favoritism toward popular subreddits.

## 5 Classification Task

### 5.1 Baseline Topic Models

To obtain baseline metrics for classification and divergence, and to survey our dataset, we implemented several popular topic models, both supervised and unsupervised.

**Unsupervised Topic Detection.** In order to verify that the 31 chosen subreddits have distinct, well-defined topics, we trained a Latent Dirichlet Analysis (LDA) [3] model to detect 31 topics (each identified by a cluster of words) from the top-level comments. The comments were clipped to 20 words for consistency with the eventual recurrent neural topic model. We also filtered out stop-words and stemmed the remaining words.

To evaluate our hypothesis of topic degeneration in deeper levels of discussions, we also trained a LDA model on comments on each of levels 1–4. These topics are compared to the topics detected in top-level comments to see if topics become less defined at deeper levels of discussions.

**Supervised Topic Classification.** Using traditional Bag-of-Words (BOW) bigram and TF-IDF [7] features, we trained two supervised topic classifiers - a support vector machine (SVM) and a multinomial Naive Bayes on top-level comments. Again, the comments were clipped to 20 words for consistency with our later model.

These classifiers are evaluated using a classification accuracy across the 31 classes. We expect the test accuracies to be higher than random chance ($1/31 = 3.23\%$). To test our hypothesis of topic drift using these classifiers, we evaluate them at different levels of discussion to obtain classification accuracies at each level. we expect a decrease in classification accuracies at deeper levels, which indicate a measurable topic drift.

### 5.2 Neural Topic Model

Our neural model is a single level, single-directional recurrent neural network using Long Short Term Memory units [4]. The final hidden state of the recurrent neural network $h_T$ is then fed as the input into a softmax classifier with a single fully-connected layer to produce probabilities $\hat{y} = \text{softmax}(Wh_T + b)$. A simple schematic is shown in Figure 1. Each $h$ node is a Long Short

Term Memory unit. Arrows represent dependencies in the forward pass, and each arrow is followed in the reverse direction in the backward pass.

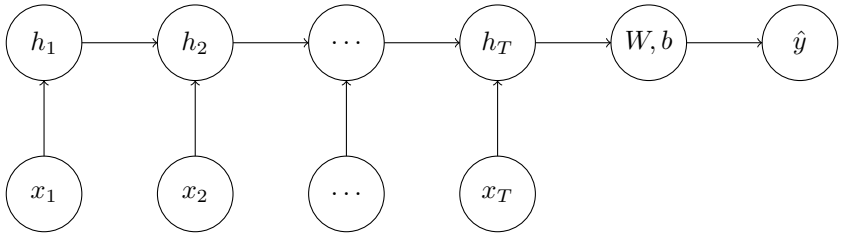



Figure 1: RNN with LSTM cells fed into softmax classifier

Our data are $(x^{(i)}, y^{(i)})$, where $x^{(i)}$ is a representation of a single comment, and $y^{(i)}$ is the one-hot representation of the true class (subreddit) the comment was taken from. For each $x^{(i)}$, the inputs to the RNN $x_1^{(i)}, \ldots, x_T^{(i)}$ are vector representations of the first $T$ words in the comment.

One forward pass calculates predicted class probabilities $\hat{y}^{(i)}$ for one $x^{(i)}$ The loss function is a regularized cross-entropy loss:

$$J = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y^{(i)} \log \hat{y}^{(i)} + \frac{1}{2}\lambda \|W\|_F,$$

where $N$ is the total number of data points, $K$ is the number of classes, and $\lambda$ is the regularization parameter. All $x_t$ and $h_t$ for $t = 1, \ldots, T$ and $W$ are updated in the backward pass.

# 6 Topic Modeling Experiments

## 6.1 Baseline Results and Evaluation

**Human performance.** We performed a test of human performance on 124 examples. Based on only the words of the comment, the human correctly guessed 43, giving a 36.7% accuracy. We recognize the difficulty of classification from low human performance on classification. Generic comments such as "I don't think you have the right idea there," or "Moderator, please remove this post. It is inappropriate for this subreddit," could belong to any subreddit. Also, many subreddits are already closely related. For example, the set of possible comments in `r/askscience` could be almost a subset of those in `r/space`. Subreddits such as `creepy` and `nosleep` are easily confused.

**Unsupervised Topic Detection.** Some topics detected by LDA in top-level comments are shown in Table 1, along with their top associated words. The topics detected were very coherent. By manual inspection, we could even pick an "associated subreddit," a subreddit in our dataset whose comments we felt would fall closely in that cluster.

| Topic ID | Top topic terms | Associated subreddit |
|---|---|---|
| 1 | Good great pretty video documentary watch movie lot music people life movies watching shows film | `r/movies` |
| 2 | Read book books reading story time good write series april writing library written great thomas | `r/books` |
| 4 | Mind brain philosophy universe theory exist sense idea state human real knowledge physical question | `r/philosophy` |
| 6 | Energy light mass field speed gravity black force magnetic space universe hole object matter star | `r/space` |
| 9 | Story season robin great episode love ted bad episodes mother character years good time series | `r/television` |

Table 1: Example topics detected among top-level comments

The topic clusters detected in level 4 comments are shown in Table 2. These topics are very general and do not have a clear corresponding subreddits, demonstrating that, at an aggregate level, topic drift has occurred by 4 levels into threads.

| Topic ID | Top topic terms | Associated subreddit |
|---|---|---|
| 1 | Post comment link read reddit edit page subreddit source search tries video check rule origin | no clear subreddit |
| 6 | Year time only change happen age before event start long 000 thousand day probably many | no clear subreddit |
| 8 | Energy water force point field heat mass air cause orbit particle speed does light wave | `r/science` or `r/space`? |
| 12 | Really people very does problem something work sound actual stuff bad lot idea anything good | no clear subreddit |
| 28 | good really something feel tries teah friend hope definitely pretty time bit figure always someone | no clear subreddit |

Table 2: Example topics detected among level 4 comments

**Supervised Topic Classification.** After training the SVM and Naive Bayes classifiers on top-level comments, they were evaluated on comments at each level. A plot of the classification accuracies at each level of conversation are shown in Figure 2. The initial classification accuracies at top-level comments (32.0% for SVM and 34.5% for Naive Bayes) are better than random chance. In addition, both models show a decreasing trend in the classification accuracy at deeper levels of conversation, demonstrating a measurable topic drift as conversations progress (at an aggregate level).

| Level | SVM | MNB |
|---|---|---|
| 0 | 32.0% | 34.5% |
| 1 | 18.4% | 21.8% |
| 2 | 18.1% | 20.3% |
| 3 | 15.1% | 18.8% |
| 4 | 14.0% | 17.4% |

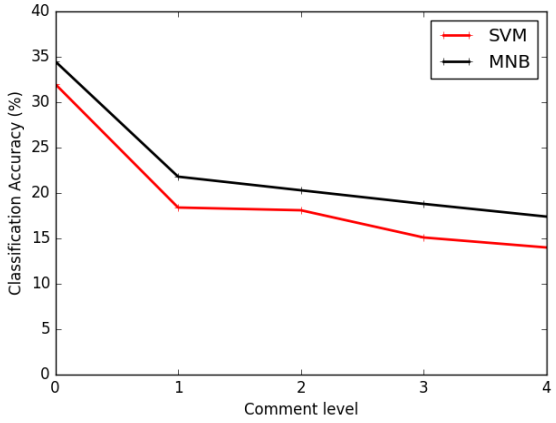


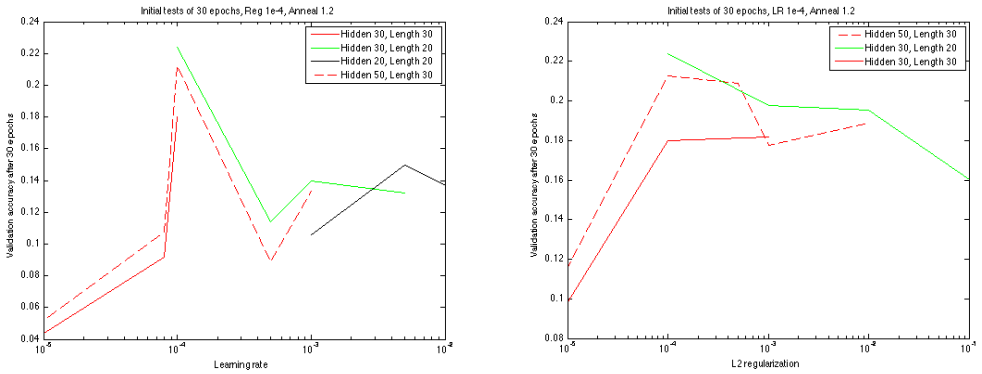

Figure 2: Baseline classification accuracies at different conversation levels.

### 6.2 Neural Network Training

**Initialization.** The hidden states of the LSTM are initialized using the Tensorflow defaults. The word vectors are initialized to be GloVe vectors [5] of 50 dimensions if they are found in the GloVe vocabulary. Otherwise, they are initialized uniformly at random to values between $-0.5$ and $0.5$.

**Hyperparameter search.** For the hyperparameter search, we conducted a total of 200–300 tests, and we varied size of the LSTM hidden state, the annealing rate (only choosing between 1.2 and 1.5, occasionally using 1.0 as a "control"), the learning rate. We also experimented with a couple of parameters for the minimum and maximum post lengths. Posts shorter than the minimum length would not be included as training data, and posts longer than the maximum length would be clipped.

We used a "small" dataset for a total of 21,700 training examples and 6,200 validation examples, again with the same number of training and validation examples from each class as the "medium" dataset, giving us the welcome property of accuracy corresponding to loss.




(a) Sample of accuracy vs. learning rate

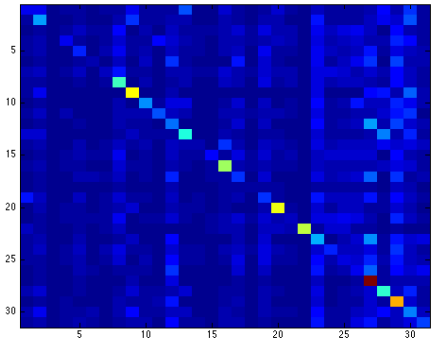
(b) Sample of accuracy vs. regularization

Figure 3: Sample plots from hyperparameter search

Figure 3 shows a small sample of the results from hyperparameter tuning from the "small" dataset. We performed an initial, larger search on the "small" dataset, then refined parameters to do more search on the "medium" dataset. The parameters we ended up using for the "medium" dataset were a hidden size of 30, a learning rate of $10^{-4}$, a L2 regularization of $10^{-4}$, an annealing rate of 1.2, and to use the first 20 words of all posts longer than 10 words.

Interestingly, 20 is a relatively small number of words to use, as comments can range into the multi-paragraph length. This suggests that top-level comments establish relevance in the first 20 words.

A confusion matrix for the final classifier is shown in Figure 4. The brown square corresponds to the subreddit `r/personalfinance`, which has a very high classification accuracy.



| id | subreddit | id | subreddit |
|---|---|---|---|
| 1 | science | 17 | tifu |
| 2 | space | 18 | UpliftingNews |
| 3 | creepy | 19 | Art |
| 4 | Documentaries | 20 | EarthPorn |
| 5 | gaming | 21 | OldSchoolCool |
| 6 | nosleep | 22 | photoshopbattles |
| 7 | sports | 23 | DIY |
| 8 | television | 24 | food |
| 9 | askscience | 25 | GetMotivated |
| 10 | books | 26 | LifeProTips |
| 11 | history | 27 | personalfinance |
| 12 | TwoXChromosomes | 28 | philosophy |
| 13 | dataisbeautiful | 29 | WritingPrompts |
| 14 | InternetIsBeautiful | 30 | Futurology |
| 15 | Jokes | 31 | gadgets |
| 16 | nottheonion | | |

Figure 4: Confusion matrix across 31 subreddits

## 7 Topic Divergence Analysis

We use the final trained classifier to study topic divergence at different levels. There are various signals we search for as we analyze divergence. We use both the classification outputs and the intermediate softmax probabilities to produce quantitative measures of divergence.

### 7.1 Aggregate-level divergence

As before, in Figure 2, we expect the classification accuracy to decrease in deeper levels of conversation. We evaluate the classification accuracy of our model on different levels of conversation, and plot it as well, in Figure 5. As expected, there is a decreasing classification accuracy at deeper level comments, demonstrating that our classifier is able to detect topic divergence at this aggregate level.

| Level | SVM | MNB | LSTM |
|---|---|---|---|
| 0 | 32.0% | 34.5% | 21.8% |
| 1 | 18.4% | 21.8% | 11.3% |
| 2 | 18.1% | 20.3% | 13.2% |
| 3 | 15.1% | 18.8% | 12.1% |
| 4 | 14.0% | 17.4% | 14.0% |

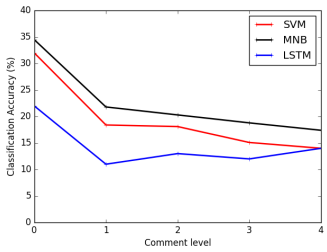
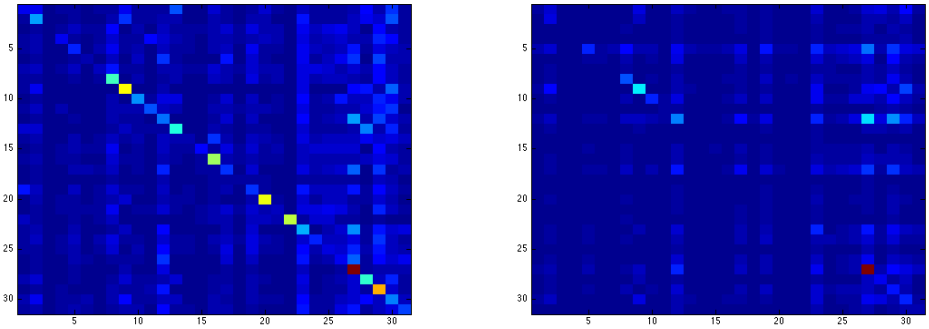



Figure 5: Classification accuracies at different conversation levels.

Another way to measure this aggregate topic divergence is to visualize the confusion matrix when our classifier is evaluated on level 4 comments, and compare it to the confusion matrix when evaluated on top-level comments in Figure 4. The confusion matrix for level 4 comments is shown in 6, with the initial confusion matrix reproduced adjacent to it for a visual comparison. As expected, the confusion matrix at level 4 shows a strong topic degeneration for most topics, except for `r/personalfinance`, which is still mostly on-topic by level 4.



(a) Top-level confusion matrix

(b) Level 4 confusion matrix

Figure 6: Confusion matrices at different-level comments

### 7.2 Thread-level divergence

To examine divergence at the thread-level, we organized our data into threads. We used only a single chain of a thread at a time, so that we were not studying a general tree structure.

We use the classifier to map each comment to a point in the subreddit topic space, and see which comment first begins to change the topic. For instance, the threads shown in Figures 7 and 8 both have topic changes occurring at level 3. Looking at the content of each comment, the predictions are correct (or reasonably correct, as in comment 2 of Figure 8). The first change in the predicted subreddit corresponds to the post that first deviated topic. Note that <UNK> represents words not in our vocabulary, and ellipsis (...) indicate where posts were truncated.

| Thread comments | prediction | KL divergence |
|---|---|---|
| [–] ah so if you find any joke distasteful you're an <UNK>. Wow. permalink embed | r/Jokes | 0.0 |
| [–] Well you don't like the joke so you probably think it's shit. Also the joke is <UNK> permalink embed parent | r/Jokes | 1.8 |
| [–] Tomatoes are made of cells. I am made of cells. Therefore I am a tomato. permalink embed parent | r/askscience | 42.4 |

Figure 7: Example from r/Jokes

| Thread comments | prediction | KL divergence |
|---|---|---|
| [–] Yeah but in Colorado we have mountains without living in Alaska. permalink embed | r/EarthPorn | 0.0 |
| [–] We have an awesome constitution that values privacy and basic human rights. I'm afraid to go to Colorado at... permalink embed parent | r/history | 5.6 |
| [–] While we're talking about things that rarely if ever happen, I'd like to not get eaten by... permalink embed parent | r/askscience | 14.0 |

Figure 8: Example from r/Jokes

In addition to looking at the predicted labels, we can even quantify how far the topic has deviated from the first top-level comment, by measuring the Kullback-Leibler (KL) divergence of each comment's softmax predictions from the top-level comment's softmax predictions. We expect comments that first topically deviated to correspond to spikes in the KL divergence. This pattern is true for both examples.

## 8 Conclusion

This project studied the dynamics of topic divergence on Reddit. Our work accomplished this study via two tasks: first, we built a topic model using word embeddings; and second, we used this classifier to quantify different levels of topic divergence. We do unfortunately note the expected but low accuracies, which were in part because of computational limitations and in part because of inherent limitations in the data, as seen in the low human accuracy.

However, even with lower classification accuracies, the neural classifier was useful for us to study topic divergence at an aggregate scale, and perform case studies on specific threads. As expected, comments identified as off-topic corresponded to spikes in KL divergence from the original top-level comment.

### 8.1 Future Work

In our project, one main limitation in the classifier was our restriction on predicting a subreddit from only the words in a comment. One extension would be to try to incorporate more context, such as looking at the parent comment, children comments, or even sibling comments. In fact, this project did not exploit the tree structure apart from simply using a chain. This would not only possibly help classification accuracy, it would also give another look at divergence in a local context.

We also acknowledge that we can extend the neural classifier model in various ways, such as making it bi-directional, or even building in components that specifically target topic prediction.

Another direction would be to examine topic divergence within comments. Suppose we have a comment that is identified as a trigger of topic drift. We can classify prefixes of the comment (i.e. at every word) to determine when the topic drift first occurs within the comment, and calculate the softmax cross-entropy losses at these steps. A spike in loss would correspond to the location in a comment where the topic changes.

Overall, there exist many promising datasets and directions for exploring topical divergence in discussion forums.

## References


[1] Backstrom, L., Kleinberg, J., Lee, L., & Danescu-Niculescu-Mizil, C. (2013, February). Characterizing and curating conversation threads: expansion, focus, volume, re-entry. *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 13-22). ACM.

[2] Blei, D.M. & McAuliffe, J.D. (2010) Supervised Topic Models. *Advances in Neural Information Processing Systems 20*.

[3] Blei, D.M., Ng, A.Y. & Jordan, M.I. (2003) Latent Dirichlet Allocation. In Lafferty, J (eds.), *Journal of Machine Learning Research 3*, pp. 993-1022. Cambridge, MA: MIT Press.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780. Chicago

[5] Pennington, J., Socher, R. & Manning, C. (2014) GloVe: Global Vectors for Word Representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[6] Reddit Comments Corpus. (2015) Archive.org. `https://archive.org/details/2015_reddit_comments_corpus`

[7] Salton, G., Wong, A. & Yang, C.S. (1975) A Vector Space Model for Automatic Indexing. *Communications of the ACM* **18**(11):613-620.

[8] Wang, L., Lui, M., Kim, S. N., Nivre, J., & Baldwin, T. (2011, July). Predicting thread discourse structure over technical web forums. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 13-25). Association for Computational Linguistics.